



**Hochschule Reutlingen**  
Reutlingen University



**INF**

Studiengang  
Medien- und  
Kommunikationsinformatik

Uwe Kloos, Natividad Martínez, Gabriela Tullius (Hrsg.)

# Informatics Inside **Interaction Design**

Informatik-Konferenz an der Hochschule Reutlingen  
06. Mai 2015

ISBN 9783000493959



9 783000 493959

# Impressum

## **Anschrift:**

Hochschule Reutlingen  
Reutlingen University  
Fakultät Informatik  
Medien- und Kommunikationsinformatik  
Alteburgstraße 150  
D-72762 Reutlingen

Telefon: +49 7121 / 271-4002

Telefax: +49 7121 / 271-4042

E-Mail: [infoinside@reutlingen-university.de](mailto:infoinside@reutlingen-university.de)

Internet: <http://www.infoinside.reutlingen-university.de>

## **Organisationskomitee:**

Prof. Dr. Gabriela Tullius, Hochschule Reutlingen

Prof. Dr. Natividad Martínez, Hochschule Reutlingen

Prof. Dr. Uwe Kloos, Hochschule Reutlingen

Christian Henssler

Jens Mahler

Matthias Merk

Greg Rauhöft

Yannik Wahn

Alexander Zimmermann



**Hochschule Reutlingen**  
Reutlingen University

Copyright: © Hochschule Reutlingen, Reutlingen 2014

Herstellung und Verlag: Hochschule Reutlingen

ISBN 978-3-00-049395-9

# Inhaltsverzeichnis

## Paper

---

**Christian Henssler**

*Techniken zur Deformation von virtuellen Menschmodellen.....* 08

**Jens Mahler**

*Der Einfluss von Scroll-Activated Animations auf Usability und User Experience..* 16

**Matthias Merk**

*Entwicklung eines ablenkungsfreien Anzeigekonzeptes für informierende Fahrerassistenzsysteme .....* 28

**Greg Rauhöft**

*Unermessliche Weiten in virtuellen Umgebungen .....* 42

**Yannik Wahn**

*Two Stream Hypothesis: Adaptionseffekte bei sozialen Interaktionen mit Avataren in Virtual Reality .....* 50

**Alexander Zimmermann**

*Mit Virtual Reality zum E-Learning - eine prototypische Anwendung .....* 58

## Shortpaper

---

**David Randler, Ralf Dauenhauer, Nils Tofahrn**

*Erweiterung der Interaktionsmöglichkeiten von Multi-Touch-Geräten.....* 68

**Philipp Kopp, Michael Grupp, Peter Poschmann, Hans-Joachim Böhme, Matthias Rätsch**

*Tracking System with Pose-Invariant Face Analysis for Human-Robot Interaction.* 70

**Steffen Witting, Uwe Kloos, Matthias Rätsch**

*Animation of Parameterized Facial Expressions for Collaborative Robots.....* 72

**Tobias Fluck, Simone Liegl, Veronika Rein, Steffen Schellig, Palina Vorobeva**

*Interaktionsgestaltung in der CryEngine 3.....* 74

**Alexander Kunz, Julian Freund, Dominic Lyons, Maksym Gaiduk**

*Technologien und Projekte des Internet of Things.....* 76

**Johannes Schirm**

*Tutorix - Ein System zur Tutorienverwaltung .....* 78

# Mit Virtual Reality zum E-Learning - eine prototypische Anwendung

Alexander Zimmermann  
Reutlingen University  
Alexander.Zimmermann@Student.  
Reutlingen-University.DE

## Abstrakt

Mit dem Kunstbegriff ‚Virtuelle Realität‘ beschreibt man die Darstellung von künstlichen Welten und die Interaktion mit den selbigen. Meist verbindet man damit teure Spiel- und Filmproduktionen. Doch durch derzeitige Entwicklungen können auch kleine Entwicklerstudios und Endanwender auf Bewegungserkennungssysteme zurückgreifen. In dieser Ausarbeitung werden zwei Prototypen vorgestellt, die auf eben diese Systeme zurückgreifen. In den Prototypen soll eine Interaktion mit der Umwelt und ein „Mittendringefühl“ im Rahmen von Serious Games ermöglicht werden.

## Schlüsselwörter

e-learning-environment, training, microsoft kinect, leap motion, oculus rift, virtual reality, motion capturing

## CR-Kategorien

H.5.2 User Interfaces - *Interaction styles*

---

Betreuer Hochschule: Prof. Dr. Uwe Kloos  
Hochschule Reutlingen  
Uwe.Kloos@Reutlingen-  
University.DE  
Betreuer Firma: Dr. Matthias Bues  
Fraunhofer IAO  
matthias.bues@iao.fraunhofer.de  
Informatics Inside 2015  
Wissenschaftliche Vertiefungskonferenz  
06. Mai 2015, Hochschule Reutlingen  
Copyright 2015 Alexander Zimmermann

## 1 Einleitung

Als Serious Games werden Spiele bezeichnet, deren hauptsächlicher Zweck nicht die Unterhaltung des Spielers ist. Beispielsweise Spiele, die zur Bildung oder der Vorstellung eines Produktes dienen. Dabei werden dennoch auf die für Spiele typischen Technologien zurückgegriffen, beispielsweise deren Entwicklungsumgebungen (genannt ‚Engine‘) oder die Interaktionsgeräte.

Für diesen Artikel entstanden zwei Prototypen, die auf die Engine Unity sowie auf die Oculus Rift VR, die Microsoft Kinect v2 und die Leap Motion zurückgreifen. Für einen Prototyp wurde eine Kombination aus Oculus Rift VR und Leap Motion verwendet, für den anderen die Oculus Rift VR sowie die Microsoft Kinect v2. Die Prototypen sollen zeigen, ob es mit den Komponenten möglich ist, Lernumgebungen zu entwickeln, und dabei einen Vergleich der Interaktionsarten schaffen.

Im folgenden Teilkapitel wird auf vorangegangene Arbeit von Studenten dieser Hochschule eingegangen. In Kapitel 2 beschäftige ich mich mit der Darstellung der Anwendung. Kapitel 3 zeigt die physischen Komponenten die für die Anwendung genutzt werden. Im vierten Kapitel werden die Prototypen dargestellt und Vor- und Nachteile der jeweiligen Interaktionskonzepte aufgezeigt. Das letzte Kapitel fasst die Erkenntnisse zusammen

und bietet einen Ausblick auf zukünftige Arbeit in diesem Feld.

Die Arbeit an diesem Artikel dient als Vorbereitung auf meine Masterarbeit. Der Zweck ist es, einen Vergleich zwischen den Interaktionsmethoden zu schaffen, um eine Entscheidung für die Thesis treffen zu können. Um feststellen zu können, wie Nutzer mit den Geräten interagieren, wäre eine Evaluation nötig. Im Rahmen meiner Masterarbeit findet eine Evaluation statt, da ich innerhalb dieser eine fertige Anwendung vorstellen möchte. Die Evaluation wird zeigen, wie Nutzer auf die eingesetzten Interaktionsmethoden reagieren.

### **1.1 Verwandte Arbeiten**

Matthias Gutekunst zeigt in [4], wie die Montage einer Tiefenkamera an der Oculus Rift VR genutzt werden kann, um die Immersion in virtuellen Umgebungen zu steigern. Er benutzt den Ansatz, das von der Kamera aufgenommene Bild der Hände des Nutzers in die virtuelle Umgebung einzubinden. Seine Anwendung schneidet dabei die Hände aus dem Umfeld aus und fügt es als Bild in die Ebene vor der virtuellen Kamera in der grafischen Umgebung ein. Die dargestellte Hand in der grafischen Umgebung ist dadurch nur ein plattes Objekt und lässt in seinem Prototyp keine Interaktion zu.

Aus dieser Anwendung ging eine Evaluation hervor, wie Personen Abstände und Größen in virtuellen Welten wahrnehmen. Gutekunst verglich dabei die Wahrnehmung, wenn die Testpersonen ihren eigenen Körper zur Orientierung sehen oder aber wenn sie keinen eigenen virtuellen Körper haben.

## **2 Anwendung**

Der Anwendungsfall für die von mir verwendeten Interaktionsmethoden ist eine Schulung von Arbeitern an Maschinen. Diese hypothetischen Maschinen sind zum Zeitpunkt der Schulung nicht zugänglich. Konkrete Arbeitsschritte wären

- das Aufheben und Anbringen eines Bauteils
- Reindrehen oder Rausdrehen von Schrauben
- Betätigung von Schaltern.

Die fertige Schulungsanwendung sollte nicht nur eine grafische Umgebung bieten, sondern auch eine Hilfestellung über die nächsten Arbeitsschritte geben. Auch ein Rückkopplungssystem ist für eine richtige E-Learning-Anwendung nötig. Weiteres darüber im Kapitel ‚Ausblick‘.

### **2.1 Zielsetzung**

Anhand des in Kapitel 2 definierten Anwendungsszenarios sollen in den Prototypen Interaktionen ermöglicht werden. Diese Interaktionen spiegeln die Arbeitsabläufe an den Industriemaschinen wider. Der virtuelle Charakter soll mit seiner Bewegung an die erfasste Person im Trackingbereich geknüpft sein. Dies soll die Immersion in der Lernwelt erhöhen. Darüber hinaus soll auf diese Art eine realitätsnahe Lernumgebung geschaffen werden.

Eine Beschränkung der Interaktionen ist in der derzeitigen Entwicklungsstufe der Technik nicht vermeidbar. Beispielsweise kann es passieren, dass der eigene Körper eine Geste mit der Hand verdeckt, sodass die stationäre Tiefenkamera diese nicht erfassen kann. Oder aber der Sensor, der an der Kamera montiert ist, kann nicht exakt deuten, ob es sich um die linke oder die rechte Hand handelt oder ob die Handinnenfläche nach oben oder nach unten zeigt. Deswegen soll das Interaktionspektrum so beschränkt werden, dass die dabei ausgeführten Bewegungen von den Kameras ‚weitestgehend eindeutig‘ erkannt werden.

## **3 Anzeige- und Interaktionsgeräte**

Für die beiden Prototypen der Anwendung wird jeweils ein Gerät für die Interaktionserkennung und zur Anzeige

verwendet. Das Oculus Rift dient bei beidem zur Anzeige, als Interaktionsgerät wird entweder die Leap Motion oder aber die Microsoft Kinect v2 verwendet.

Die Interaktionsgeräte, auch Gestensteuerungen genannt, sollen dazu dienen, in Spielen für eine höhere Immersion zu sorgen oder ein neues Konzept der Interaktion zu ermöglichen. So gibt es Anwendungen, in denen Autos durch das Drehen eines ‚Luft-Lenkrads‘ gesteuert werden können oder Tanzspiele, bei denen die Bewegung von einer Tiefenkamera erkannt, und von der Anwendung bewertet wird.

### 3.1 Oculus Rift

Das Oculus Rift (kurz: OVR, für Oculus Virtual Reality, siehe Abbildung 3) ist ein Anzeigegerät, welches man wie eine Brille auf dem Kopf trägt. Diese Art von Anzeigegerät bezeichnet man als ‚Head-mounted Display‘, kurz HMD. Das OVR besitzt eine Auflösung von 960 mal 1080 Pixel pro Auge, eine Bildwiederholrate von 75 Hertz und deckt einen Blickwinkel von 100° ab. Das normale, menschliche Sichtfeld deckt mit 176-180° in der horizontalen ein weiteres Feld ab.

Als Bildübertragungsstandard verwendet das OVR HDMI 1,4b. Die Anzeige bezieht ihren Strom über den USB-Anschluss vom Rechner.

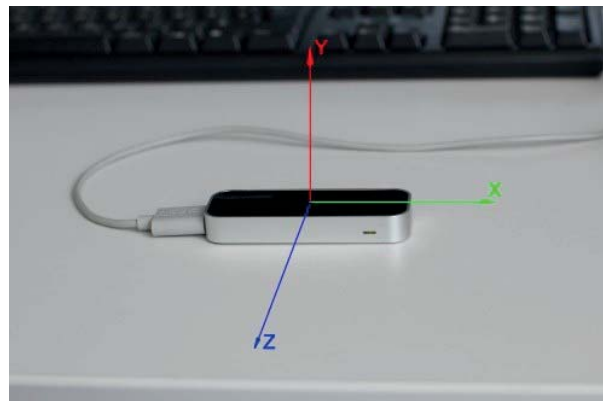
Das OVR besitzt Sensoren, die es möglich machen, durch Erkennen des Umfeldes Neigung und Rotation des HMDs zu bestimmen. So kann die Blickrichtung und die Kopfbewegung in die virtuelle Welt übertragen werden.

Zusätzlich ist im Umfang des OVR eine Positionsbestimmungskamera enthalten. Diese dient, ähnlich wie die Microsoft Kinect oder die Leap Motion, der Tiefenerkennung und kann ebenfalls als Bewegungserkennung eingesetzt werden. Allerdings ist ihr eigentlicher Einsatzzweck,

die Position des OVR im Raum zu bestimmen (siehe [1]).

### 3.2 Leap Motion

Die Leap Motion ist ein Interaktionsgerät, welches ursprünglich auf die Oberfläche eines Tisches gelegt wird. Mit einer Tiefenkamera wird dann der Bereich über dem Gerät abgetastet. So können über dem Gerät Gesten ausgeführt werden, die von der Software der Leap Motion verstanden werden. Auch räumliche Position und Orientierung der Hand und ihrer Finger können an die Computer übertragen werden. Dabei kann die Software zwischen linker und rechter Hand unterscheiden und sie gleichzeitig darstellen.



**Abbildung 1: Leap Motion mit Koordinatenachsen**

Die Leap Motion ist 13 auf 13 auf 76mm groß und tastet einen kegelförmigen Bereich mit der Höhe von bis zu 25cm ab. Der Winkel des Kegels beträgt bis zu 150° (siehe [2]).

Für die Anwendung, die in diesem Projekt entstanden ist, wurde die Leap Motion an dem OVR angebracht und dient als tragbare Bewegungserkennung für die Hände. Anders als bei dem von der Firma ursprünglich angedachten Bedienszenario, ist man auf diese Weise nicht mehr an eine Position im Raum gebunden.



### 3.3 *Microsoft Kinect v2*

Ähnlich wie der Tiefensensor des OVR, ist die Microsoft Kinect v2 eine Positionsbestimmungskamera. Es ist ein markerloses Bewegungserkennungsgerät und kann die Position einzelner Gegenstände, Hände oder gar ganzer Körper an die Computer übertragen.



**Abbildung 2: Microsoft Kinect v2**

Die Tiefenkamera der Kinect besitzt eine Auflösung von 512 mal 424 Pixel und tastet mit einer Rate von 30 Hz ab. Das Sichtfeld hat einen Winkel von 70° horizontal und 60° vertikal. Der Abtastbereich fängt 50cm vor der Kamera an und hört 450cm von der Kamera entfernt auf. Als 3D-Scannverfahren wird die „time of flight“-Technik, kurz TOF, verwendet. Hierbei wird ein Lichtimpuls ausgesandt und gemessen, wie lange der Impuls braucht, um wieder zur Kamera zurück geworfen zu werden (siehe [3]).

## 4 Prototypen

Zur Erstellung der Prototypen wurde Unity 3D in seiner Version 4.6.3 verwendet. Unity ist eine Grafikengine, die mit einem freien und mit einem kostenpflichtigen Lizenzmodell verfügbar ist. Während das OVR und die Leap Motion mit der freien Version von Unity funktionsfähig sind, ist für die Kinect v2 zwingend die kostenpflichtige Version nötig. So sind die für die Einbindung nötigen „Plugins“ erst mit der Unity3D Pro Version benutzbar.

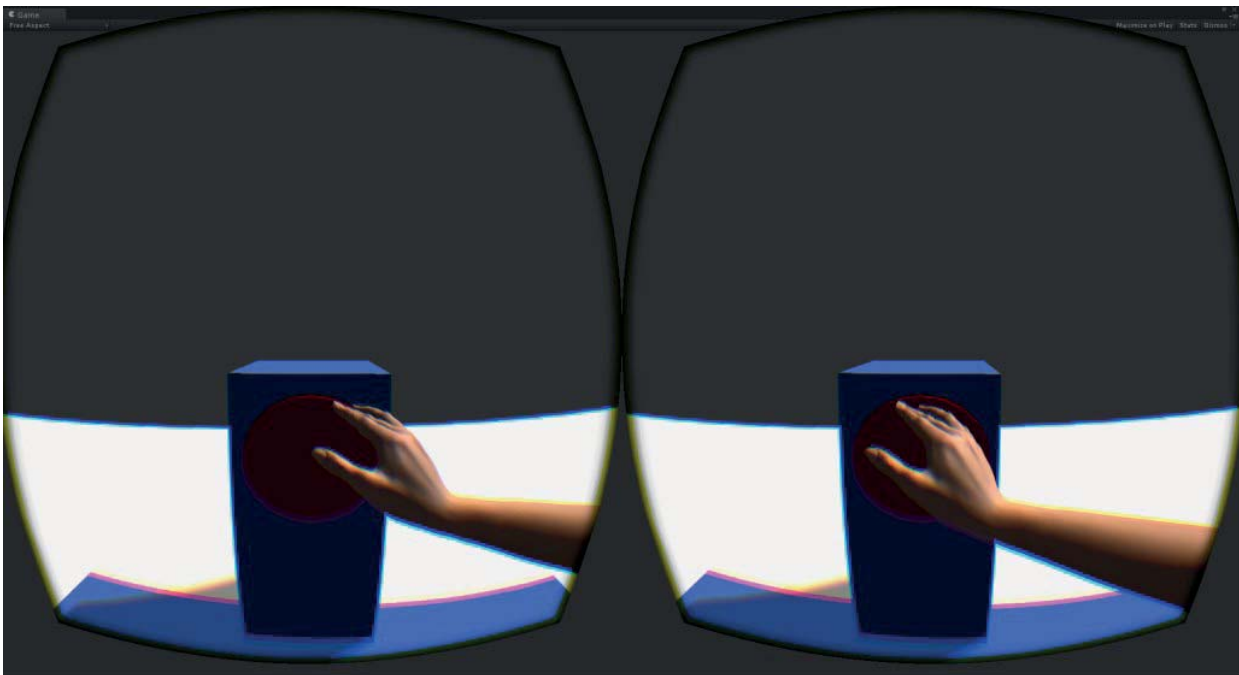
Die in Kapitel 2 vordefinierten Aktionen galt es in den Prototypen widerzuspiegeln. Es wurde eine Welt mit verschiedenen Stationen erstellt. An jeder Station konnte eine andere Aktion erprobt werden. So gibt es eine Station, von der ein Objekt aufgehoben und auf eine andere Fläche abgelegt werden muss. Zwischen dem Aufheben und Ablegen gibt es eine Distanz zu überbrücken, bei der das virtuelle Objekt gehalten werden muss. Als Objekt dient hierbei ein einfacher Würfel. An der nächsten Station ist ein Schalter auf Hüfthöhe positioniert. Diesen gilt es zu betätigen (siehe Abbildung 4). Die dritte Station enthält einen Stonehenge-artigen Aufbau, der die Kollision von Objekten miteinander demonstriert. Station 4 besteht aus einem weiteren Würfelobjekt. Dieser Würfel muss in eine dafür vorgesehene Öffnung geschoben werden. Bei der letzten Station muss eine Kugel aus einem Container gehoben und in einen anderen Container platziert werden. Die letzten beiden Stationen sind, so simpel sie sich auch anhören, die schwierigsten.



**Abbildung 3: Leap+OVR**

### 4.1 *Leap + OVR*

Der Aufbau mit der Leap Motion und dem OVR (siehe Abbildung 3) besteht auf dem ersten Blick mit seinem genauen Handtracking. Jeder Finger kann einzeln bewegt und erfasst werden und der Sensor erkennt ebenfalls, ob es sich um die linke oder rechte Hand handelt. Dadurch, dass bei



**Abbildung 4: Sicht durch OVR. Interaktion mit Schalter.**

diesem Ansatz die Positionsbestimmungskamera des OVR nicht mit verwendet wurde, entfällt allerdings das Tracking des restlichen Körpers der zu trackenden Person.

Eine Ausrichtung des Nutzers in der Welt ist nur durch einen Betreuer oder aber unter Einsatz eines Gamepads bzw. einer Maus und Tastatur möglich. Selbst ob die Person sich duckt, ist nicht in diesem Aufbau erkennbar. Dieses Problem steht aber der Tatsache gegenüber, dass man nicht an einen Bereich gebunden ist. Man könnte das OVR und die Leap Motion an ein Notebook anschließen, auf dem auch die Anwendung ausgeführt wird, und dieses Notebook dann in einen Tornister unterbringen, den der Nutzer auf dem Rücken trägt. Auf diese Art wäre man unabhängig von einem Trackingbereich.

#### 4.1.1 Qualität des Trackings

Betrachtet man die Qualität des Trackings, so werden schnell gravierende Einschränkungen sichtbar. Bewegungen wie Klatschen werden nicht wiedererkennbar in die virtuelle Welt übertragen. Hält man die Hände zusammen, mit den Handflächen zueinander zeigend, so zerfällt das Tracking komplett. Platziert man die Fingerspitzen

aufeinander, so schauen sie in der virtuellen Welt aneinander vorbei oder liegen ineinander. Auch werden Gesten oft von der eigenen Hand verdeckt. Der geringe Trackingbereich der Leap Motion macht Aktionen, die weit entfernt vom Körper liegen, unmöglich. Der maximale Abstand, der von der Leap Motion erkannt wird, wird vom Hersteller auf 25cm angegeben. Der Abstand, in dem eine Geste aber fehlerfrei erkannt wird, ist geringer. Auch erkennt die Leap Motion oftmals Pinch-Gesten, also das Drücken von Daumen und Zeigefinger aufeinander, wo keine sind.

Die Autoren in [6] geben an, dass sich die Genauigkeit des Trackings je nach Position des zu trackenden Objekts stark unterscheidet. In ihrer Versuchsreihe positionieren sie die Leap Motion auf einer Tischoberfläche, mit dem Sensor nach oben schauend. Aus dieser Position ergeben sich die verschiedenen Achsen so, dass die Y-Achse orthogonal von der Leap Motion nach oben und unten reicht. Die X-Achse verläuft horizontal durch die Leap Motion, wobei der negative Bereich links und der positive Bereich rechts von der Leap Motion liegt. Die Z-Achse ist zur trackenden Person hin positiv und nach hinten weg negativ (siehe



Abbildung 1). Guna et al. beschreiben in [6], dass die Leap Motion sich im linken Bereich, also im Bereich mit negativen X-Wert, genauer verhält als im rechten Bereich. Optimal sei aber die Mitte (bis zu 100mm Abweichung von  $X = 0$ ). Für den Z-Bereich ergab sich, dass das Tracking vor dem Sensor, also mit positiven Z-Wert, genauer ist als hinter der Leap Motion. Überträgt man dies auf den Prototypen in dieser Ausarbeitung, so bedeutet dies, dass das Tracking unterhalb der montierten Leap Motion besser ist als oberhalb. Gesten, die also über dem Kopf ausgeführt werden, werden weniger genau erkannt. Die Autoren des Artikels geben einen optimalen Z-Wert von bis zu 160mm in positiver Richtung an (siehe [6] S. 3716ff).

#### 4.1.2 Interaktion mit der Welt

An der ersten Station macht sich der Nachteil des fehlenden Positiontrackings bemerkbar. Der Würfel lässt sich nicht ohne Einsatz von Tastatur oder Hilfe von außerhalb auf die entfernte Plattform legen. Das Aufheben und Fallenlassen des Würfels ist allerdings gut und problemlos möglich. Durch das von Leap Motion mitgelieferte ‚Magnetic Pinch‘-Script wird der Würfel von der Hand angezogen, sobald diese eine Pinch-Geste vollführt. Verdeckt man aber die Geste vor der Kamera, zum Beispiel durch Drehen der Hand, sodass die Handinnenfläche von der Leap Motion weg zeigt, fällt der Würfel wieder aus der Hand.

Die zweite Station ist da hingegen einfach zu erledigen. Wenn das Handobjekt in der virtuellen Welt den Knopf der Station berührt, wird ein Event ausgelöst, welches zur Ausführung von anderen Aktionen innerhalb der Welt verwendet werden kann. Nötig ist dazu nur eine richtige Ausrichtung an dem Schalter, sodass dieser erreichbar ist.

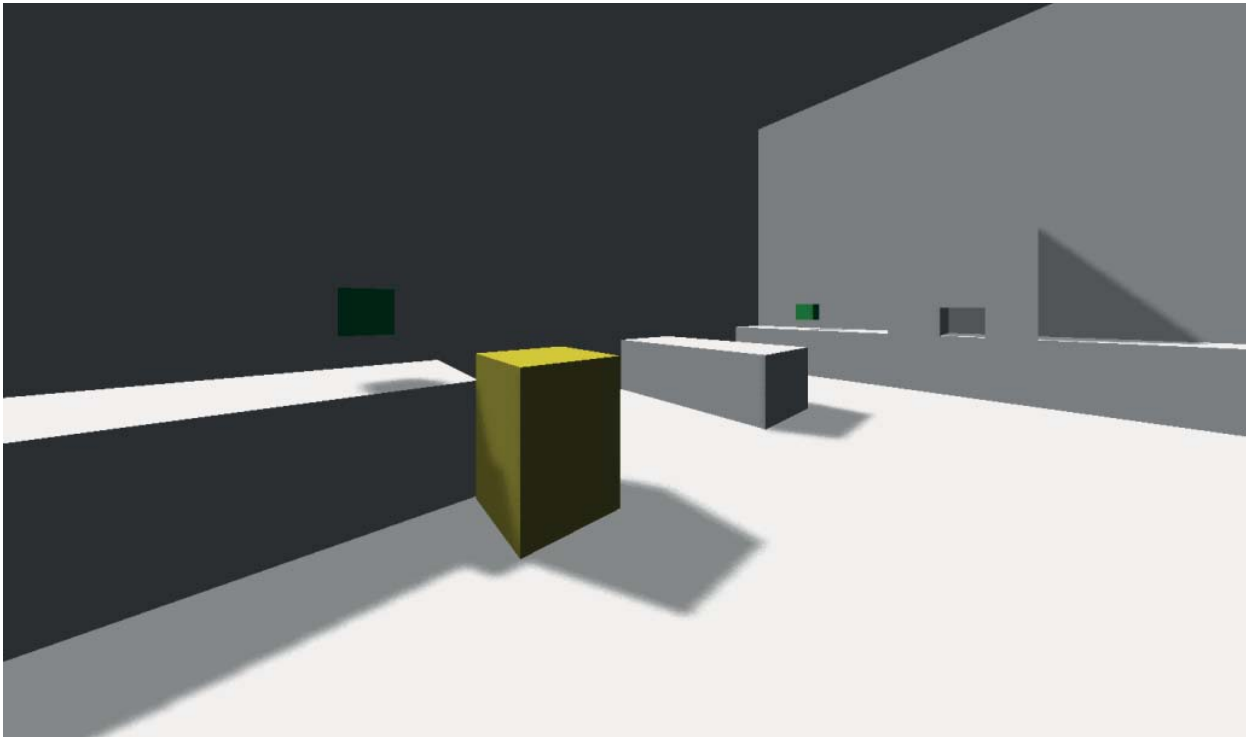
Die dritte Station, die aufgestapelten Quader, demonstriert wie gut man einzelne Objekte heben und sogar mit beiden Händen greifen und rotieren kann. Auch das Verschieben anderer Objekte durch ein gehaltenes Objekt ist möglich.

Das Einsetzen des Würfels in die Öffnung an Station 4 zeigt, dass die Darstellung des eigenen Körpers in einer virtuellen Welt stark bei der Interaktion mit der selbigen hilft. Problematisch bei ersten Iterationen der Prototypen war nur, dass sich das gehaltene Objekt weiter rotiert, obwohl es von einer Hand gehalten wird. Ein Einsetzen war deswegen nur schwer möglich, da sich das Objekt nicht berechenbar verhalten hat. Eine Kopplung der Rotation eines gehaltenen Objektes an die Handtransformation macht die Verwendung dieser Aktion für spätere Anwendungen möglich. Auch hier machen sich Verdeckungsprobleme bemerkbar. Zudem verdeckt der in der Hand gehaltene Kubus die Sicht auf die Öffnung. Der Sehbereich über das OVR ist unnatürlich klein, getragene Dinge im peripheren Blickfeld zu halten ist mit der Leap als Eingabegerät nicht möglich. Zum einen ist der als ‚peripheres Blickfeld‘ bezeichnete Bereich kleiner als man es gewohnt ist, zum anderen verliert die Leap das Tracking, sobald man aus dem geringen Bereich verschwindet.

Die letzte Station ist durch fehlendes Positiontracking unnötig schwer. Da man sich nicht ducken kann, ist das Aufheben der Kugel aus dem Container nur schwer möglich. Das gezielte Platzieren der Kugel in dem anderen Container aber nahezu unmöglich. Man muss die Kugel in der richtigen Position fallen lassen und aufpassen, dass die Leap Motion nicht einen willkürlichen Pinch erkennt, sodass die Kugel nicht von dem ‚Magnetic Pinch‘-Script zurück zur Hand gebracht wird.

## 4.2 Kinect V2 + OVR

Der erste offensichtliche Vorteil der Kinect gegenüber der Leap ergibt sich aus dem Positiontracking. Der Nutzer kann, dadurch dass er von der Kinect-Tiefenkamera erfasst wird, sich selbst in der virtuellen Welt ausrichten. Der Hersteller des Motion-Capture-Geräts gibt einen Trackingbereich von 0,5 bis 4,5 Metern an (siehe Kapitel 3.3). Khoshelham et al. zeigen in [5] aber,



**Abbildung 3: Station 1 (vorne) und Station 4 (hinten)**

dass der Trackingbereich kleiner ist, je mehr Anspruch man auf Genauigkeit legt. Sie erklären auf Basis ihrer Messungen mit der Kinect 1, dass bei maximaler Entfernung von 4,5m ein Abstand von 5,5cm zwischen den einzelnen Tiefenpunkten liegt (siehe [5] S. 14ff). Dies bezieht sich zwar auf eine niedrigere Auflösung von 320 mal 240 Tiefenbildpunkten, verteilt sich dafür aber auch auf eine geringere Sichtweite von 57° horizontal und 43° vertikal (im Vergleich die Kinect v2 512 mal 424 Bildpunkte, 70° horizontal, 60° vertikal, siehe [8]). Fürntratt et al. empfehlen in [7] deswegen, dass die Kinect v2 für das bestmögliche Ergebnis auf den Boden gestellt wird und in einem 25° Winkel entgegen dem Nutzer ausgerichtet wird. Zudem sollte die zu trackende Person in einem Abstand von 2,3m positioniert sein. Dadurch kann der Ungenauigkeit und der zitterhaften Sprünge einzelner virtueller Knochen entgegengewirkt werden (siehe [7] S. 4).

Ein weiteres Merkmal ist, dass die Kinect aus einer anderen Perspektive aufnimmt. Steht der Tiefensensor vor einer Person, so nimmt er Gesten, die von der Person weg gerichtet sind, besser auf als die Leap

Motion. Jedoch sind die zur Person hin und von der Kinect v2 weg gerichteten Gesten durch die eigene Hand verdeckt. Zudem ergeben sich Verdeckungsprobleme, sobald man sich von der Kamera weg dreht.

Mit der Kinect ist man zwar ortsgebunden an dem von der Kinect abgedeckten Trackingbereich, allerdings ist dadurch eine Umpositionierung im virtuellen Raum möglich. Wie bei der Leap auch, müssen große Positionsänderungen aber mit einer Eingabe über Tastatur und Maus bzw. Gamepad durchgeführt werden.

#### 4.2.1 Qualität des Trackings

Klatschbewegungen werden, gleich wie bei dem Leap+OVR-Aufbau, auch hier nicht wiedererkennbar übertragen. Presst man die Hände aufeinander, so ergibt es aber ein sichtbar besseres Ergebnis. Die Verbesserung ist erklärbar dadurch, dass nicht nur die einzelnen Finger bzw. Hände, sondern auch die Arme als Orientierung für das Handtracking genutzt werden können.

Das Kinect SDK hält drei vorgefertigte Zustände für die Handgestenerkennung bereit. Eine geöffnete Hand, eine geballte Faust und die Hand mit abgespreiztem

Zeigefinger und Daumen. Durch die Unterscheidung von geöffneter und geschlossener Hand können Greifinteraktionen vollführt werden.

Mit der Kinect lassen sich ganze Körper bestimmen. Für den angedachten Anwendungsfall ist dies allerdings nicht nötig. Zudem unterscheidet die Kinect bei den Händen nur zwischen Hand bzw. Handfläche, Fingerspitze und Daumen. Ein knochengenaues Tracking wie bei der Leap Motion gibt es hier nicht. Anders als die Leap Motion ist das von der Kinect erzeugte Skelett in dem Prototyp sehr robust. Einzig die Handknochen erzeugen ein Zittern, welches in den Skripten ausgeglichen werden muss.

#### 4.2.2 Interaktion mit der Welt

Die Aufgabe, einen Würfel aufzunehmen und auf eine etwas entfernte Plattform zu legen, ist dank des Positionstrackings ohne andere Eingabehilfsmittel möglich. Vorausgesetzt, man ist zum Start der Aktion so im Trackingbereich positioniert, dass man zum Ende hin nicht aus diesem hinaus läuft.

Das Auslösen von Schaltern ist ebenso simpel wie bei dem Leap+OVR-Aufbau. Auffallend ist, dass man keine haptische Rückkopplung darüber erhält, ob der Schalter betätigt wurde. Aktionen, die aus dem gedrückten Schalter resultieren, werden durch Events ausgelöst. Diese Events werden aufgerufen, sobald sich ein vordefiniertes Objekt, in diesem Fall die Fingerspitze, durch den Kollisionsbereich des Schalters bewegt.

Die Stonehenge-Station sowie das Einsetzen des Würfels in der Öffnung erweisen sich ebenfalls als ähnlich durchführbar wie mit der Leap Motion. Vorteilhaft ist hier aber, dass man nicht gezwungen ist, auf das gegriffene Objekt hinab zu schauen, um die Hand nicht aus dem Trackingbereich zu verlieren. Dies ergibt sich daraus, dass der Trackingbereich bei dem Kinect+OVR-Aufbau nicht an die Ausrichtung des OVR gebunden ist. Mangels peripheren Sichtbereichs erhält man allerdings keine Information darüber, ob sich das eben in der Hand gehaltene Objekt immer noch in der Hand befindet.



Abbildung 4: Station 3 (Mitte) und Station 5 (rechts)

Die letzte Station, an welcher es gilt, eine Kugel aus einem Eimer zu entnehmen und sie in den anderen Eimer zu legen, ist durch das Positionstracking einfacher. Durch hinknien kann der Abstand zu den Eimern verringert werden. Dadurch ist es einfacher abzuschätzen, ob die Kugel in dem Eimer landet, oder aber man kann die Kugel in den Eimer legen.

## 5 Schlussfolgerung

Die Leap Motion bietet ein sehr genaues Fingertracking, während die Kinect die Hand nur sehr grob erfasst. Für den Anwendungsfall mit dem OVR ist eine Steigerung der Immersion zwar wünschenswert, durch die mangelnde Robustheit des Leap Motion-Trackings gilt es dennoch abzuwägen. Für die Anwendung einer Lernumgebung für Maschinen ist fotorealistische Grafik ein weniger relevantes Kriterium. Zudem lässt sich durch gutes Modeling, Rigging und Skinning eines Charakters eine realistischere Handbewegung erzeugen als im Prototyp.

Die Kombination aus Leap und OVR ist in Bezug auf die Bewegungsfreiheit im Vorteil. Eine anderweitige Positionierung im Raum ist bei beiden Prototypen nötig. Allerdings ergibt sich der Gewinn an Bewegungsfreiheit nur dadurch, dass das Interaktions- und das Anzeigegerät an ein Notebook angeschlossen ist, welches in einem Tornister untergebracht wurde. Der Performanz der Darstellung von virtuellen Welten kommt dies nicht zu Gute. Die Nutzung in diesem rechenleistungsschwachen Aufbau schließen die Entwickler des OVR derzeit aber indirekt aus. Sie empfehlen, Anwendungen mit 75 Bildern pro Sekunde oder höher darzustellen.<sup>1</sup> Grund hierfür ist, die schnell aufkommende Übelkeitsempfindung bei Verwendung von Head-mounted-Displays bestmöglich zu reduzieren.

---

<sup>1</sup> <https://support.oculus.com/hc/en-us/articles/201835987-Oculus-Rift-Development-Kit-2-FAQ>

## 5.1 Ausblick

In den hier vorgestellten Prototypen wurde das Tracking direkt aus dem Sensor ausgelesen und in der Anwendung verwendet. Das Ergebnis des Trackings könnte aber mit einem verarbeitenden Zwischenschritt verbessert werden. Wie bereits in Kapitel 2.1 angedeutet, handelt es sich bei virtueller Realität auch teilweise um eine ungenaue Wissenschaft. Man könnte, statt die Position in jedem Frame auf den virtuellen Körper zu übertragen, den Übergang von einem Frame auf den nächsten Frame darstellen. Kombiniert mit einem Dämpfungsfaktor, der drastische Sprünge von Knochenpositionen in einzelnen Frames oder kurzen Framesequenzen reduziert. So könnte man für ein robuster-erscheinendes Ergebnis sorgen.

Alternativ könnte man auch, ähnlich wie bei der Erstellung von animierbaren Charakteren, unnatürliche Knochenpositionen definieren und vorbeugen. Sodass die Anwendung erkennt, dass eine Knochenposition nicht wahrscheinlich ist und dafür eine wahrscheinlichere einnimmt.

Im Ausblick auf die eigene Arbeit ist zu entscheiden, welche Kombination von Interaktions- und Anzeigegeräten für eine Lernanwendung am besten ist. Für meine Masterthesis soll die Steuerung und Interaktion der Welt noch mit einem System verwoben werden, welches den Fortschritt einer Aufgabe und die Richtigkeit der Lösung ausgibt. Dieses System kann beispielsweise der konstruktivistischen Lerntheorie folgen und am Ende den Nutzer für seine Arbeit bewerten.

## 6 Literaturverzeichnis

- [1] Oculus Rift VR Datenblatt. Website, 2015. Online verfügbar unter <https://www.oculus.com/dk2/>; Besucht am 1.3.2015.
- [2] Leap Motion Datenblatt. Website, 2015. Online verfügbar unter

- <https://www.leapmotion.com/product>; Besucht am 1.3.2015.
- [3] Microsoft Kinect v2 Datenblatt. Website, 2015. Online verfügbar unter <http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx>; Besucht am 1.3.2015.
- [4] M. Gutekunst: Augmented Virtuality zur Steigerung der Immersion in virtuellen Umgebungen. Human-Centered-Computing. Informatics Inside 2014, Tagungsband. ISBN 978-3-00-045427-1
- [5] K. Khoshelham, S. O. Elberink: Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. Sensors 2012; DOI: 10.3390/s120201437
- [6] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, J. Sodnik: An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. Sensors 2014; DOI: 10.3390/s140203702
- [7] H. Fürntratt, H. Neuschmied: Evaluating Pointing Accuracy on Kinect V2 Sensor. Proceedings of the 2<sup>nd</sup> International Conference on Human-Computer Interaction. Prag, Tschechien. 2014.
- [8] M. Szymczyk: How does the Kinect 2 compare to the Kinect 1. Website, 2014. Online verfügbar unter <http://zugara.com/how-does-the-kinect-2-compare-to-the-kinect-1>; Besucht am 3.3.2015